

REMARKS

Claims 36 – 48 have been amended to recite a tangible computer readable medium. No claims have been added or cancelled. Therefore, claims 1 – 48 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 102(e) Rejection:

The Office Action rejected claims 1-11, 14-24, 27-33 and 36-46 under 35 U.S.C. § 102(e) as being anticipated by Bass et al. (U.S. Patent 6,549,956) (hereinafter “Bass”). Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 1, contrary to the Examiner’s assertion, Bass clearly fails to disclose receiving a message in a data representation language sent to a client platform in the distributed computing environment from a service in the distributed computing environment, wherein the message includes a data representation language representation of an event generated by the service. Bass teaches a mechanism for connecting disparate publication and subscribe domains via the Internet in which two channel adapters together act as a bridge across the network. Specifically, Bass teaches the use of existing network protocols (SMTP, TCP/IP, etc) via existing holes in firewalls (column 2, lines 32-34). The Examiner cites portions of Bass (col. 2, lines 4-9, and 15-31, and column 3, lines 43-50) that describe how Bass’ channel adapters receive published events, translate them into a message format suitable for transmission over a network, and send them to a channel adapter on another platform that translates the message back into the original event information.

The Examiner contends that by disclosing the translation of event information into network protocol messages, Bass discloses “that an event (message) can be represented in any data representation language and will be converted back into the event format for use in the other domain” (Office Action, page 3, lines 4-6). The Examiner

further argues that by stating that channel adapters can convert event information into a format acceptable by the network, Bass discloses, "that an event (message) can be represented in any data representation language." However, the Examiner's interpretation of Bass is incorrect. The Examiner is arguing that the phrase "a format acceptable by the network" discloses the use of any data representation language. However, without some clear teaching by Bass regarding the use of a data representation language, Bass cannot be said to anticipate a message including a data representation language representation of an event. Furthermore, without some clear teaching by Bass, the Examiner's speculation regarding Bass' use of data representation languages for messages is clearly improper in a rejection under 35 U.S.C. § 102(e) based on anticipation.

In response to the above argument, the Examiner, in the Response to Argument, refers to Bass' teachings regarding sending event information in an email via SMTP. Apparently the Examiner is arguing that SMTP is a data representation language. However, SMTP is a protocol, not a data representation language. Data representation languages are well understood in the art. No one of ordinary skill in the art would consider SMTP (or any other similar network protocol) to be a data representation language.

Bass teaches only the translation of event information into existing network protocol messages, such as an SMTP email, TCP/IP packet, or FTP transfer message. Bass does not teach that the messages sent using these protocols are messages in a data representation language. Bass teaches the use of existing network protocols in order to take advantage of the fact that existing network protocols use existing holes in firewalls and other security mechanisms (see, column 2, lines 15-35). For instance, Bass teaches that an event is formatted for transmission on a network (such as the Internet) and that "[t]he format may use transmission control protocol/Internet protocol (TCP/IP), simple mail transport protocol (SMTP), File Transfer Protocol (FTP), or whatever protocol is useable by the connecting network" (column 3, lines 35-42). Bass does not mention using messages in a data representation language. Nor is there any reason to use

messages in a data representation language in Bass's system, since none of the existing communications protocols advocated by Bass inherently use messages in a data representation language. Data representation languages are *specific types of languages* traditionally used in the prior art to describe documents or other content. The prior art does not teach the use of a data representation language to represents events in messages between entities in a distributed computing environment.

In response the above argument, the Examiner merely objects to Applicants' use of the phrase "data representation language messages" to refer to messages in a data representation language, arguing that Applicants' claims do not recite the phrase "data representation language messages." Thus, the Examiner does not actually provide any rebuttal of Applicants' argument regarding the fact that Bass fails to disclose messages in a data representation language.

Additionally, Bass fails to anticipate wherein the message includes a data representation language representation of an event generated by the service. In contrast, Bass teaches channel adapters that "convert the event information into a format acceptable by the network" (column 2, lines 15-18). The "format acceptable by the network" in Bass is not described as a data representation language representation of an event. Bass clearly does not teach a message that includes a data representation language representation of an event. The Examiner cites only col. 2, lines 4-9, and 15-31 of Bass that, as noted above, describe how a channel adapter translates event information into network protocol messages. The Examiner does not cite any portion of Bass that refers to any data representation language representation of an event.

Furthermore, Bass fails to anticipate sending the data representation language representation of the event to one or more processes registered to receive the event from the service. The Examiner cites column 2, lines 9-15, where Bass describes a process adapter subscribing to and receiving an event via a channel adapter. However, Bass only teaches that the process adapter receives the event, not a data representation language representation of the event. The Examiner argues that by teaching how a channel adapter

reformats an event for transmission over the Internet, Bass discloses the use of a data representation language representation of events. The Examiner's interpretation of Bass is incorrect. As noted above, Bass teaches only translating event information into a format suitable for transmission over the Internet via any of a number exiting network protocols (such as TCP/IP, SMTP, FTP, etc). However, nowhere does Bass mention that the event information is a data representation language representation of an event.

In response the Applicants' argument above, the Examiner responds by citing Bass column 3, lines 43-50 where Bass describes how channel adapters receive published events, translate them into a message format suitable for transmission over a network, and send them to a channel adapter on another platform that translates the message back into the original event information. Thus, when Bass' channel adapter delivers the event to the subscribing process, the channel adapter has already, "re-transformed" the email (used to send the event information) back into the event (See, Bass, column 3, lines 45-50). Bass specifically states that the channel adapter delivers the re-constituted event, *rather than any data representation language representation of the event*, to subscribing processes.

Applicants respectfully remind the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Bass clearly fails to anticipate receiving a message in a data representation language sent to a client platform in the distributed computing environment from a service in the distributed computing environment, wherein the message includes a data representation language representation of an event generated by the service.

For at least the reasons given above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Arguments similar to those presented above regarding claim 1 apply to claims 14 and 36 as well.

Regarding claim 2, Bass fails to anticipate receiving a data representation language schema on the client platform, wherein the data representation language schema defines a message interface for a set of events generated by the service. The Examiner cites column 3, lines 43-50 that describes Bass' channel adapters. Bass teaches that each channel adapter includes two interfaces, a framework interface and a protocol interface (column 3, lines 53-64). The framework interface includes domain specific protocols for communicating published and subscribed events with a domain broker. The protocol interface includes network specific protocols that enable the adapter to couple with the Internet. The Examiner has not cited any portion of Bass that teaches a *data representation language schema* defining a message interface for a set of events. Instead, Basses Bass teaches that each channel adapter includes two different interfaces for communicating event information.

The Examiner also cites column 2, lines 4-15 and column 4, line 43 – column 5, line 15 where Bass teaches how each of his channel adapters are configured with a set of events it will export to a peer adapter in another domain. Bass teaches how a system administrator configures each channel adapter to receive and transmit specific events and how channel adapters exchange, or export, lists of events that they will be communicating. The Examiner argues that exchanging event lists amounts to receiving a data representation language schema defining a message interface for a set of events. However, Bass' event list exchange only informs the channel adapter which events will be communicated. Bass does not teach that his event export lists make up a data representation language schema. Bass also does not mention that the event export lists are exchanged using a data representation language. Furthermore, Bass does not describe his event export lists as defining message interfaces. To the contrary, as discussed above, Bass teaches (column 3, lines 53-64) that each channel adapter includes a protocol

interface that includes network protocol messages. A channel adapter's protocol interface has nothing to do with the list of events that it may send and receive. Bass teaches that the channel adapter can convert any event into an appropriate network protocol messages. Thus, the exchange of exported event lists cited by the Examiner does not teach anything regarding receiving a data representation language schema defining a message interface.

Additionally, Bass does not teach generating an event message endpoint for the client platform according to the data representation language schema. The Examiner cites Bass' teachings regarding the receiving of events listed on an event type list (column 4, line 43 – column 5, line 15) and argues that Bass discloses generating an event message endpoint according to a data representation language schema by describing how an event on an event type list is received and re-published via the channel adapter. The Examiner's interpretation of Bass is clearly incorrect. As discussed above, Bass' exported event type lists are not data representation language schemas. Moreover, not only do the event type lists in Bass not involve the generation of any message endpoints, they also have absolutely nothing to do with a data representation language schema. Thus, Bass clearly fails to disclose generating an event message endpoint for the client platform according to the data representation language schema.

In the Response to Arguments, the Examiner responds to the arguments above by merely repeating the rejection of claim 2. The Examiner does not provide any additional explanation or argument regarding Applicants' arguments. Thus, the Examiner has not provided any actual rebuttal to Applicants' arguments.

Thus, for at least the reasons give above, the Examiner's rejection of claim 2 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 15 and 37 as well.

Regarding claim 3, Bass fails to disclose the event message endpoint subscribing to one or more of the set of events generated by the service, wherein the service is configured to send messages including data representation language representations of an event to subscribers to the event when the event is generated. The Examiner cites column 3, lines 43-50 of Bass describing how an event is delivered to a subscribing channel adapter. However, as discussed above regarding claims 1 and 2, Bass fails to teach anything regarding a service configured to send messages including *data representation language representations of events*. Instead, Bass teaches that events are converted into network protocol messages, such as SMTP email messages, for transmission over the Internet where they are converted back into the original event information for republishing in a different domain. However, as described above regarding claims 1 and 2, such network protocols are not data representation languages. Bass does not describe these protocol messages as being messages in a data representation language. Furthermore, nowhere does Bass mention a service configured to send messages including data representation language representations of an event.

In response to the above arguments, the Examiner, in the Response to Arguments, cites column 3, lines 49-50 where Bass states that channel adapters then delivers the event to any subscribing process adapters within the domain. However, as noted above, prior to delivering the event to the subscribing process adapters, the channel adapter converts the event information from the network protocol message (e.g. SMTP email message) back into the event. (see, Bass, column 3, lines 45-50). The event delivered by the channel adapters is clearly not a data representation language representation of the event. Thus, the Examiner's cited passage actually supports Applicants' argument.

For at least the reasons above, the rejection of claim 3 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claim 38 as well.

In regards to claim 4, Bass fails to disclose wherein the data representation language message from the service includes an authentication credential for the service. Bass additionally fails to disclose the event message endpoint using the authentication credential for the service to authenticate the data representation language message as being from the service. The Examiner cites column 4, line 57 to column 5, line 15 of Bass that describes how Bass' channel adapters are configured to send and receive various events. Please see the discussion above regarding claim 2 for a more detailed discussion of this portion of Bass. The Examiner does not provide any argument or discussion regarding how Bass' exported event type lists have relevance to a data representation language message including an authentication credential. Nowhere does Bass mention anything regarding data representation language messages including authentication credentials nor about event message endpoints using an authentication credential to authenticate the data representation language message.

In the Response to Arguments, the Examiner cites column 1, lines 56-60 of Bass and argues that Bass' stated purpose for his invention, namely, "there is a need in the art for a mechanism to link to disparate PUB/SUB domains together without compromising security, reducing performance, be easy to implement, and still allow for information transfer between the two domains" discloses the specific limitations of Applicants' claim 4. However, a general statement regarding Bass intention that his invention no compromise security does not disclose or anticipate the specific limitation of a data representation language message from a service including an authentication credential for the service. Nor does the cited passage anticipate an event message endpoint using the authentication credential for the service to authenticate the data representation language message as being from the service.

As noted above regarding claim 1, anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. The passages cited by the Examiner can in no way be considered to disclose each and every element of Applicants' claim 4, arranged as in the claim. Thus, for at least the reasons above, the rejection of claim 4 is not supported by

the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 20, 33 and 39 as well.

Regarding claim 5, Bass fails to disclose the event message endpoint verifying type correctness of the data representation language message according to the data representation language schema. The Examiner cites column 2, lines 24-27 and column 3, lines 45-50 of Bass. However, the first cited portion only describes how Bass' channel adapters use a plurality of states and status messages to track and indicate the delivery, receipt, and publication of events. The second cited portion describes how an event is transformed into an email message via SMTP and then re-transformed back into the event upon receipt.

Neither of the Examiner's cited portions of Bass has anything to do with *verifying type correctness* of a data representation language message *according to a data representation language schema*. The various states and status messages indicating delivery, receipt, and publication of events only track and help to guarantee that event messages are eventually delivered to the subscribing process adapter. These states have nothing to do with verifying type correctness of a data representation language message according to a data representation language schema. Similarly, converting an event into an email message via SMTP (or another network protocol message) and converting the message back into an event has nothing to do with verifying type correctness of a data representation language message according to a data representation language schema. In fact, nowhere does Bass make any reference whatsoever to verifying type correctness of a data representation language message according to a data representation language schema.

Furthermore, the Examiner has argued that Bass' exported event type lists constitute a data representation language schema (see, Final Office Action, pages 14-15, regarding claim 2). However, Bass does not teach that an exported event type list has anything do with the states and status messages indicating delivery, receipt, and

publication of events or have anything to do with converting events into SMTP messages. Thus, the Examiner's interpretation of Bass is inconsistent and thus cannot be correct.

In the Response to Arguments, the Examiner responds to the above arguments by cited column 4, lines 18-24 where Bass describes how each channel adapter includes a reporting mechanism. Bass describes this reporting mechanism has informing an administrator of the status of events and that the administrator can "determine if there are any events that are stuck, and the state in which they are stuck." However, reporting on the state of events as they flow through Bass' system does not disclose the specific functionality of verify type correctness of a data representation language message according to a data representation language schema. Not only does the cited passage fail to mention verifying type correctness of any messages, the passage also fail to mention anything regarding the use of a data representation language schema to verify type correctness.

Therefore, the rejection of claim 5 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 16 and 40 as well.

Regarding claim 6, Bass fails to anticipate wherein the data representation language schema defines a set of messages that the service may send to the event message endpoint and further fails to teach the event message endpoint verifying the correctness of the data representation language message from the service according to the data representation language schema. The Examiner once again cites column 2, lines 24-27 and column 3, lines 45-50 of Bass. However, as noted above regarding claim 5, the first cited portion only describes how Bass' channel adapters use a plurality of states and status messages to indicate the delivery, receipt, and publication of events and the second cited portion describes how an event is transformed into an email message via SMTP and then re-transformed back into the event upon receipt. As discussed above regarding claim 5 (for which the Examiner cites the same portions of Bass), neither of the

Examiner's cited portions have anything to do with a data representation language schema defining a set of messages that a service may send to an event message endpoint. Additionally, neither of the cited passages mentions an event message endpoint verifying the correctness of a data representation language message according to the data representation language schema.

Specifically, the various states and status messages (indicating delivery, receipt, and publication of events) only help to track and guarantee that event messages are eventually received by the subscribing process adapter. These states have nothing to do with verifying the correctness of a data representation language message according to a data representation language schema. Similarly, converting an event into an email message via SMTP (or another network protocol message) and converting the message back into an event is not verifying type correctness of a data representation language message according to a data representation language schema.

Thus, the rejection of claim 6 is clearly not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 17 and 41 as well.

Regarding claim 8, Bass fails to disclose each of the one or more processes providing an event handler callback method to the event message endpoint. The Examiner cites column 4, lines 57-60. However, this portion of Bass only teaches that Bass' channel adapters subscribe to and publish events, but fails to describe any mechanism for delivering the events other than via existing network protocols. Nowhere does Bass teach providing an event handler callback method to an event message endpoint.

In response to the arguments above, the Examiner cites column 4, line 43 through column 5, line 16 and refers to Bass' teaching that channel adapters republish events to interested process adapters. Specifically the Examiner refers to Bass teaching how "prior

to transfer of events between the domains, the respective process and channel adapters of the domains must be configured to send and receive the different events” (Bass, column 4, lines 57-59). However, merely stating that the process and channel adapters must be configured to send and receive events in no way discloses, teaches, or even implies providing an event handler callback method to an event message endpoint. Without some clear and specific teaching by Bass regarding providing an event handler callback method, the Examiner is merely speculating as to the details of Bass system.

Bass further fails to teach the event message endpoint calling an event handler method of each process registered with the event message endpoint and the event message endpoint passing the data representation language representation of the event to each called event handler. The Examiner cites column 3, lines 22-50 where Bass describes how his channel adapters convert events to and from network protocol message and how events are sent over the Internet and re-published in other domains. However, nowhere does Bass describe an event message endpoint calling an event handler method. Nor does Bass teach an event message endpoint passing a data representation language representation of an event to each called event handler. The Examiner merely states, “the reference teaches that the processes as well as the adapters are configured to do the claimed element” and further claims, “the c[h]annel adapters are capable of executing the task as claimed.” However, without any supporting teaching from Bass, the Examiner’s rejection amounts to nothing more than mere hindsight speculation and conclusory statements.

Thus, for at least the reasons given above, the rejection of claim 8 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 22 and 43 as well.

Regarding claim 10, Bass does not disclose receiving the data representation language schema of the service in a service advertisement of the service. The Examiner cites column 2, lines 4-15, column 3, lines 43-50 and column 4, line 43 – column 5, line

15, where Bass describes how his channel adapters are configured with a set of events it will export to its peer adapter in another domain. Please refer to the remarks above regarding claim 2 for a discussion of these portions of Bass. The Examiner apparently contends that Bass's use of exported event type lists include service advertisements. However, the exported event type lists have absolutely nothing to do with a service advertisement that includes a data representation language schema.

In the Response to Arguments, the Examiner again cites column 2, lines 4-15, column 3, lines 43-50 and column 4, line 43 – column 5, line 15 of Bass, without providing any additional argument or interpretation regarding Applicants' argument above.

For at least the reasons given above, the rejection of claim 10 is not supported by the prior art and removal thereof is respectfully requested. Similar arguments as those presented above apply to claims 18 and 45 as well.

Regarding claim 27, Bass fails to anticipate a service process configured to generate a message in a data representation language. The Examiner cites column 2, lines 49, and 15-31 of Bass and argues that converting event information into a format acceptable by the network discloses that an event (message) can be represented in any data representation language. The Examiner's interpretation of Bass is incorrect. As discussed above regarding claim 1, Bass teaches the use of existing network protocols such as SMTP, TCP/IP, or FTP which have absolutely no bearing whatsoever on the use of a data representation language. Bass does not mention anything regarding using data representation language messages.

Bass further fails to anticipate wherein the message includes a data representation language representation of the event generated by the service process. The Examiner does not cite any passage in Bass that refers to a message including a data representation language representation of an event, as suggested by the Examiner. Instead, Bass teaches

that the event information is translated into a network protocol message, as described above regarding claim 1. Furthermore, as defined in the art, existing network protocols do not include data representation language representation of events.

Bass also does not anticipate wherein each of the one or more event message gate units is operable to distribute the data representation language representation of the event, as asserted by the Examiner. Also as noted above regarding claim 1, Bass teaches that once received by a channel adapter, the network protocol message is converted back into the original event information. Thus, in order to distribute data representation language representations of an event, an event would have to originally be a data representation language representation of the event. However, Bass does not teach anything regarding data representation language representations of events. The Examiner has not cited any passage of Bass that refers to data representation language representations of an event.

In the Response to Arguments, the Examiner cites elements 16 and 17 of Bass' FIG. 1 and refers to the previous Response to Arguments regarding claim 1. However, FIG. 1 of Bass does not illustrate a message in a data representation language or data representation language representations of events generated by the service process. Furthermore, as noted above, Bass teaches the use of existing network protocols, such as SMTP, TCP/IP, or FTP, which, as discussed above, are not data representation languages.

For at least the reasons given above, the rejection of claim 27 is not supported by the prior art and removal thereof is respectfully requested.

Regarding claim 29, Bass fails to anticipate a service process configured to provide a data representation language schema defining a message interface for a set of events generated by the service and also fails to teach wherein one or more event message gate units are generated according to the data representation language schema. The Examiner cites column 3, lines 43-50 that describes Bass' channel adapters. Bass teaches that each channel adapter includes two interfaces, a framework interface and a

protocol interface (column 3, lines 53-64). The framework interface includes domain specific protocols for communicating published and subscribed events with a domain broker. The protocol interface includes network specific protocols that enable the adapter to couple with the Internet. The Examiner has not cited any portion of Bass that teaches a data representation language schema defining a message interface for a set of events. Instead, Bass teaches that each channel adapter includes two different interfaces for communicating event information.

The Examiner also cites and column 2, lines 4-15 and column 4, line 43 – column 5, line 15 where Bass teaches how his channel adapters are configured with a set of events it will export to its peer adapter in another domain. Bass teaches how an administrator configured each channel adapter to receive and transmit specific events and how channel adapters exchange lists of events that they will be communicating. The Examiner argues that exchanging event lists amounts to receiving a data representation language schema defining a message interface for a set of events. However, Bass' event list exchange only informs the channel adapter which events will be communicated. Bass does not teach that his event export lists are data representation language schemas. Bass does not mention that the event export lists are exchanged using a data representation language. Furthermore, Bass does not describe his event export lists as defining a message interfaces. On the contrary, as discussed above, Bass teaches (column 3, lines 53-64) that each channel adapter includes a protocol interface that includes network protocol messages. A channel adapter's protocol interface has nothing to do with the list of events that it may send and receive. Bass teaches that the channel adapter can convert any event into an appropriate network protocol messages. Thus, the exchange of exported event lists cited by the Examiner does not teach anything regarding receiving a data representation language schema defining a message interface.

Bass also fails to teach generating event message gate units according to a data representation language schema. The Examiner cites Bass' teachings regarding the receiving of events listed on an event type list (column 4, line 43 – column 5, line 15) and argues that Bass discloses generating event message gate units according to a data

representation language schema by describing how an event on an event type list is received and re-published via the channel adapter. The Examiner's interpretation of Bass is clearly incorrect. As discussed above, Bass' exported event type lists are not data representation language schemas. Not only do the event type lists fail to define any message interfaces, they also do not use a data representation language.

Thus, for at least the reasons given above, the Examiner's rejection of claim 29 is not supported by the prior art and removal thereof is respectfully requested.

Regarding claim 31, Bass does not teach a service process configured to provide the data representation language schema in a service advertisement. The Examiner cites column 2, lines 4-15, column 3, lines 43-50 and column 4, line 43 – column 5, line 15, where Bass describes how his channel adapters are configured with a set of events it will export to its peer adapter in another domain. Please refer to the remarks above regarding claim 2 for a discussion of these portions of Bass. The Examiner apparently contends that Bass use of exported event type lists include service advertisements. However, the exported event type lists have absolutely nothing to do with a service advertisement that includes a data representation language schema.

Thus, for at least the reasons given above, the rejection of claim 31 is not supported by the prior art and removal thereof is respectfully requested.

Regarding claim 32, Bass fails to teach the event message endpoint subscribing to one or more of the set of events generated by the service, wherein the service is configured to send messages including data representation language representations of an event to subscribers to the event when the event is generated. The Examiner cites column 3, lines 43-50 of Bass describing how an event is delivered to a subscribing channel adapter. However, as discussed above, Bass fails to teach anything regarding a service configured to send message including data representation language representations of an

event. Instead, Bass teaches that events are converted into network protocol messages for transmission over the internet where they are converted back into the original event information for re-publishing in a different domain. Nowhere does Bass mention a service sending messages including data representation language representations of an event.

Thus, the rejection of claim 32 is not supported by the prior art and removal thereof is respectfully requested.

Section 103(a) Rejection:

The Office Action rejected claims 12, 13, 25, 26, 34, 35, 47 and 48 under 35 U.S.C. § 103(a) as being anticipated by Bass in view of Meltzer et al. (U.S. Patent 6,542,912) (hereinafter "Meltzer"). Applicants respectfully traverse this rejection for at least the reasons given above regarding their respective independent claims.

In regard to both the section 102 and 103 rejections, Applicants also assert that the rejections of numerous ones of the dependent claims are further unsupported by the cited art. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-65700/RCK.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☐ Petition for Extension of Time
- ☐ Notice of Change of Address
- ☐ Other:

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: July 13, 2005